

How to setup MIIC development environment

For new developers, learn how to setup the MIIC webserver build & test environment. These instructions were written with mac OS X El Capitan in mind, but it all the tools we use are cross-platform.

Required Software

First you need to install the following required software:

- Maven 3.0.4 or better. On OS X I use the "homebrew" package manager and then "brew install maven".
- Java JDK 1.8 or newer
- Eclipse Spring Tool Suite. Grab the latest from <https://spring.io/tools> and run "check for updates".
- PostgreSQL 9.2 or newer.
 - On the mac I use postgres.app (<http://postgresapp.com>) because it installs & runs very easily.
 - You should also install PGAdmin (<https://www.pgadmin.org/download/>) to help with database configuration steps

Install SGP4 Jar to Maven

The MIIC project uses Maven to build. Maven is able to download all MIIC project dependencies *except* for sgp4, because this does not exist on any Maven repository that I have been able to find.

Therefore, you must checkout the sgp4 jar from the repository and manually add them to your local Maven cache.

First, download the jar from the repository:

- `source:trunk/MIIC_Client/thirdparty/sgp4/sgp4.jar`

Next, add them to your maven repository:

```
% mvn install:install-file -Dfile=<path-to-sgp4.jar> -DgroupId=sgp4  
-DartifactId=sgp4 -Dversion=1 -Dpackaging=jar
```

Checkout & build MIIC Project

- Clone the MIIC git repository from <https://git.earthdata.nasa.gov/scm/mii/multi-instrument-intercalibration.git>
 - Use the master branch unless you have a reason to use otherwise
 - This requires a NASA ECE account that is authorized to use git. You might need to consult with ECE admin [Ross Bagwell](#) for help.
- In Eclipse, select Import->Projects from Git.
 - Use "existing local repository" and add the location of the MIIC repo.
 - Choose "import existing Eclipse projects" since eclipse files are stored in the repo.
 - Select these projects: MIIC (parent project), miic-plugins, miic-server, plugin-api, plugin-driver, rest-client
- Build the projects. Everything builds cleanly, right? 😊

Fixing Build Errors

New checkouts typically do not build cleanly inside Eclipse, at least not right away. This is due to problems with Maven<->Eclipse integration and some of the Maven plugins that we are using.

Don't panic.

The first thing to note is that maven is the build tool of record, not eclipse. Go to the shell and run maven from the MIIC folder:

```

LASLA40053623:MIIC abartle$ pwd
/Users/abartle/MIIC/multi-instrument-intercalibration/MIIC_Client/workspac
e/MIIC

LASLA40053623:MIIC abartle$ mvn install
...
[INFO]
-----
[INFO] Reactor Summary:
[INFO]
[INFO] miic-project ..... SUCCESS [
0.188 s]
[INFO] plugin-api ..... SUCCESS [
3.947 s]
[INFO] miic-plugins ..... SUCCESS [
0.587 s]
[INFO] miic-server ..... SUCCESS [02:18
min]
[INFO] rest-client ..... SUCCESS [
1.333 s]
[INFO] plugin-driver ..... SUCCESS [
0.255 s]
[INFO]
-----
[INFO] BUILD SUCCESS
[INFO]
-----
[INFO] Total time: 02:24 min
[INFO] Finished at: 2017-01-25T12:53:31-05:00
[INFO] Final Memory: 89M/870M
[INFO]
-----

```

You should see a message like above, with all projects building cleanly.

If you restart Eclipse, you will now see that this command created a MIIC web archive:

- ../multi-instrument-intercalibration/MIIC_Client/workspace/MIIC/server/target/miic.war

This is what you need to deploy to the Tomcat server to run MIIC.

You may also find that incremental build-and-deploy steps may start working after having built from the shell.

Create Tomcat Server

To run inside Eclipse you need a tomcat 7.x server for testing.

- File->new->Server
- Select "Pivotal tc Server v2.5 - v2.9". Click next.
- If the installation directory is blank, this needs to be updated to the Spring Tool Suite's folder containing servers.
 - e.g. /Users/abartle/sts-bundle/pivotal-tc-server-developer-3.2.2.RELEASE
- Ensure runtime environment has "workbench default JRE" and "tomcat-7.0.39.A.RELEASE" as the version. (OK if newer version 7)
- Click next. Create new instance

- Name the instance "tomcat7" and use template "base"
- Create the server.

You should now be able to open your "Servers" view (Window->Show view servers) and see your "tomcat7" server.

Create MIIC Database

Start pgAdmin. You may need to add a connection to the database running at "localhost".

Create a database named "miic".

MIIC Configuration

At minimum you will need a PostgreSQL database named "miic" and some configuration files need to be updated. See here for more details: [How to deploy MIIC web archive](#).

Setting properties

You will need to customize properties found in the files:

- `.../multi-instrument-intercalibration/MIIC_Client/workspace/MIIC/server/src/main/resources/miic.properties`
- `.../multi-instrument-intercalibration/MIIC_Client/workspace/MIIC/server/src/main/resources/miic.jdbc.properties`

The values to change are pretty obvious:

- URL to OPeNDAP server (e.g. <https://dev-miic.larc.nasa.gov/opensdap>)
- Path to folders the server uses for cache
- Login information for the database if required

Since these are your custom settings, don't edit the ones in the repository. Instead, copy them to your server's conf folder and make the changes there.

You can find your server's conf folder by rt-clicking on your server "tomcat7", "show deployment location". Unless you placed it somewhere different it will be in the Spring Tool Suite installation folder, e.g.:

`.../sts-bundle/pivotal-tc-server-developer-3.2.2.RELEASE/tomcat7/conf`

Deploy and Test

1. Deploy the miic.war to the server. You can open the "servers" window, right-click your server, add and remove, then add miic-server.
2. Start the server. You can also run the server in debug to debug MIIC code.
3. Point your browser to localhost:8080/miic and you should see the MIIC welcome page!

Startup timeout

If the server times out when starting, double click the server, you should see a "Timeouts" section with start timeout. Set this to something larger.

Add Yourself as a User

The MIIC server will allow URS logins but the MIIC web app doesn't allow anyone in unless they are also in the MIIC database. We did this to retain control over who is using the server.

- Start pgAdmin. Open the miic database and find the table "user_accounts".
- If you tried to log in with URS, you should already have an entry in this table.
- Change the role "ROLE_PROVISIONAL" to "ROLE_ADMIN" or "ROLE_USER".

Related articles

- [How to setup MIIC development environment](#)